Sparse Classifier Fusion for Speaker Verification

Ville Hautamäki, Tomi Kinnunen, Filip Sedlák, Kong Aik Lee, Bin Ma, and Haizhou Li *Senior Member, IEEE*

***Abstract*—State-of-the-art speaker verification systems take advantage of a number of complementary base classifiers by fusing them to arrive at reliable verification decisions. In speaker verification, fusion is typically implemented as a weighted linear combination of the base classifier scores, where the combination weights are estimated using a logistic regression model. An alternative way for fusion is to use classifier ensemble selection, which can be seen as sparse regularization applied to logistic regression. Even though score fusion has been extensively studied in speaker verification, classifier ensemble selection is much less studied. In this study, we extensively study an integrated process of classifier selection and fusion over a collection of twelve I4U spectral subsystems on the NIST 2008 and 2010 speaker recognition evaluation (SRE) corpora.**

***Index Terms*—Classifier ensemble selection, linear fusion, speaker verification, experimentation**

I. Introduction

SPEAKER verification is the task of accepting or rejecting an identity claim based on a person's speech sample [1]. Modern speaker verification systems utilize ensembles of *base classifiers* to arrive at an accurate verification decision by *classifier fusion*. The base classifiers might utilize, for instance, different speech parameterizations (e.g. spectral, prosodic or high-level features), classifiers (e.g. Gaussian mixture models [2] or support vector machines [3]) or channel compensation techniques (e.g. joint factor analysis [4] or nuisance attribute projection [5]).

In this study, we consider weighted linear combinations of the base classifier scores to implement fusion. With a small number of adjustable parameters, linear fusion scheme often shows good generalization performance. But it is crucial for the weights to be optimized using robust method which tolerates reasonable deviations in the base classifier score distributions. In speaker verification, the scores may vary considerably between the training and runtime data due to differences in acoustic environments and transmission channels. The obvious weight optimization strategy, minimizing error rate on the training set, easily overfits. This is because error counting, corresponding to the 0-1 *loss function* [6] of the linear classifier, defines a non-differentiable and non-convex error surface having multiple local minima.

A natural solution is to use a convex surrogate loss function instead that serves as an upper bound to the 0-1 loss function. Optimizing an upper bound will also reduce the classification error rate on the unseen data while strict convexity ensures the existence of a unique global minimum. Well-known loss functions with these desiderata include the *hinge loss* used in training SVMs [7] and the logistic loss, also known as *logistic regression* [8]. The latter one has been found reliable in independent studies of fusion in speaker verification [8], [9], [10]. Considered as the *de facto* standard in speaker verification studies, with readily available implementations (e.g. [11], [12]), we take the logistic regression model as our baseline. One further advantage of the model is that the fused scores have an interpretation as automatically calibrated *log-likelihood ratios* (LLRs). In addition to producing interpretable scores, this enables designing the verification threshold using the standard Bayes' minimum risk classifier design [13] based on known class priors and pre-specified misclassification costs.

Logistic regression is a probabilistic model of the decision boundary between two classes and its parameters (weights) are usually found as the *maximum likelihood* (ML) estimate on a training set [7]. ML solution easily overfits when the number of training scores (trials) is low relative to the dimensionality (number of base classifiers) or when the scores show large dataset-to-dataset variations. In statistical vocabulary, the ML-trained weights have large *variance* over different instances of the training set which manifests itself as fusion weights with large magnitude; consequently, even a small change in the base classifier outputs causes large change in the fusion score leading to unreliable decisions.

Motivated by this observation, we consider *regularized* [14] logistic regression whereby weight vectors with large norm are penalized. Regularization defines a constrained optimization problem where one finds a compromise between training data accuracy while avoiding weights with large magnitude. Importantly, regularized solution can also be viewed as *maximum a posteriori* (MAP) estimate whereby one imposes a prior distribution over the weights [14]. As in any practical Bayesian learning method, two additional design concerns are now introduced: (1) choosing the regularizer (functional form of the weight prior) and (2) training its parameters (regularization parameters) that act as hyperparameters. To exemplify, *ridge regression* [14] or squared Euclidean norm regularization corresponds to choosing an isotropic Gaussian prior with zero mean where the variance parameter determines the degree of regularization applied. In this study, we train the regularization parameters using a held-out validation dataset and focus on the first design question, the choice of the regularizer.

In this paper, we advocate integrating *sparse* regularization to logistic regression model training in speaker verification. This means that, rather than optimizing fusion weights for the full classifier ensemble, we would like to implement simultaneous classifier ensemble selection and fusion. There

V. Hautamäki, T. Kinnunen and F. Sedlák are with the Univ. of Eastern Finland, Joensuu, Finland (email: {villeh,tkinnu,fsedlak}@cs.joensuu.fi). K.A. Lee, B. Ma and H. Li are with Inst. for Infocomm Research (I2R), Singapore (email: {kalee, mabin, hli}@i2r.a-star.edu.sg)

The work of F. Sedlák was supported by Nanyang Technological Univ. (NTU), Singapore, works of T. Kinnunen and V. Hautamäki by Academy of Finland and the work of H. Li by Nokia foundation.

Manuscript submitted 03.01.2011, revised 26.06.2011

are several arguments favoring such approach. Firstly, even though the full system may consist of up to a dozen of base classifiers (e.g. [15]), these are often redundant; they might utilize only slightly different spectral front-ends, training parameters, acoustic models and development corpora. It is therefore reasonable to assume that the effective number of base classifiers contributing further uncertainty reduction in the fused score is relatively small. An experimental validation for this hypothesis comes from our recent study [16]. Applying exhaustive classifier selection and weight optimization from a pool of 12 classifiers [15], we found that a classifier ensemble with only 4 or 5 classifiers outperformed full ensemble in accuracy. Secondly, reducing the effective number of model parameters is expected to improve generalization performance because of reduced model variance [17]. Finally, from a practical point of view, classifier selection would lead to computationally more feasible system as well.

Even though joint selection of the classifier ensemble and training the fusion weights is a combinatorial optimization problem, it can be mathematically formulated as ℓ_0-regularization [14] where the regularizer (zeroth norm) counts the number of non-zero weights, corresponding to the selected classifier ensemble. Unfortunately, its time complexity is exponential with respect to the number of base classifiers. The usual workaround is to use ℓ_1-regularization instead, a method known as LASSO (*least absolute shrinkage and selection operator*) [18]. LASSO shrinks all the coefficients, with some of them forced to be exactly zero. By regularizing logistic regression with the LASSO constraint, we can simultaneously optimize fusion weights and perform classifier selection. Convex combination of ridge regression and LASSO leads to another regularization technique known as *elastic-net* (E-net) [19], which retains the the zeroing capability of LASSO, but because of ridge term it does not push base classifier weights to zero as aggressively as LASSO or classifier ensemble selection.

The contributions of the present study are summarized as follows. We propose that fusion device, implemented as a weight vector, is necessarily sparse. Subset of all classifiers will then bring best performance. As a practical implementation, we propose to use LASSO regularized logistic regression, to overcome the computational burden.

II. Linear Score Fusion in Speaker Verification

A. Problem Setup

We assume that, during the development phase, one has access to a development set $\mathcal{D} = \{(\mathbf{s}_i, y_i), i = 1, 2, \ldots, N_{\mathrm{dev}}\}$ containing N_{dev} score vectors from L base classifiers, $\mathbf{s}_i \in \mathbb{R}^L$. Here, $y_i \in \{0, 1\}$ indicates whether the corresponding speech sample originates from a target speaker ($y_i = 1$) or from a non-target ($y_i = 0$). Though not always the case during the NIST SRE campaigns, here we assume that these labels contain no errors. We consider linear score fusion of the form $f_{\mathbf{w}}(\mathbf{s}) = w_0 + \sum_{l=1}^{L} w_l s_l = \mathbf{w}^{\mathsf{T}}\mathbf{s}$, where $\mathbf{w} = (w_0, w_1, \ldots, w_L)^{\mathsf{T}}$ contains the classifier weights $w_1, \ldots, w_L$ (discrimination component) and the bias w_0 (calibration component). The augmented score vector $\mathbf{s} = (1, s_1, s_2, \ldots, s_L)^{\mathsf{T}}$ contains constant 1 and the base classifier output scores.

Our goal is to find the optimal weight vector (say, $\mathbf{w}^*$) so that classification errors are minimized on unseen evaluation data. Therefore, let us first introduce our classification cost function used in evaluation. Here we adopt the *detection cost function* (DCF) commonly used in the NIST speaker recognition evaluations[1] to assess the accuracy of any speaker verification system:

$$\mathrm{DCF}(\theta) = C_{\mathrm{miss}} P_{\mathrm{miss}}(\theta) P_{\mathrm{tar}} + C_{\mathrm{fa}} P_{\mathrm{fa}}(\theta)(1 - P_{\mathrm{tar}}). \quad (1)$$

Here, $P_{\mathrm{miss}}(\theta)$ and $P_{\mathrm{fa}}(\theta)$ are the miss and false alarm probabilities as a function of the decision threshold θ, P_{tar} is the prior probability of a target (true) speaker, C_{miss} is the cost of a miss (false rejection) and C_{fa} is the cost of a false alarm (false acceptance). As an example, in the core condition of the latest NIST SRE 2010 evaluation, these were fixed as $C_{\mathrm{miss}} = C_{\mathrm{fa}} = 1$ and $P_{\mathrm{tar}} = 0.001$.

In speaker verification, (1) is used for computing both the *actual* (ActDCF) and *minimum* (MinDCF) values. The actual cost refers to the DCF value obtained whenever the decision threshold θ is fixed to a particular value beforehand, whereas MinDCF indicates the oracle value (minimum) on the test set that can easily be found by linear search over the range of θ. Therefore, by definition ActDCF $\geq$ MinDCF, and the difference ActDCF - MinDCF can be used as a measure of calibration error.

B. Logistic regression

To train the fusion device, in theory one can optimize (1) directly, for instance by using a neural network [20]. For the reasons discussed above, we optimize the weights using a convex loss function instead. *Logistic regression* is a probabilistic linear model, which is based on the fact that posterior probability of the class label being the target class can be written as $p(y = 1|\mathbf{s}) = (1 + \exp\{-g(\mathbf{s})\})^{-1}$ for *any* class-conditional densities [21]. The function $g(\mathbf{s})$ takes the form $\mathbf{w}^{\mathsf{T}}\mathbf{s}$ when the class-conditional densities follow exponential family of distributions with a shared dispersion parameter (e.g. variance). We can thus express the target class posterior as $p(y = 1|\mathbf{s}) = (1 + \exp\{-(\mathbf{w}^{\mathsf{T}}\mathbf{s})\})^{-1} = \sigma(\mathbf{w}^{\mathsf{T}}\mathbf{s})$ [21], where $\sigma(.)$ is a logistic sigmoid function. The posterior for the non-target class is then $p(y = 0|\mathbf{s}) = 1 - p(y = 1|\mathbf{s}) = \sigma(-\mathbf{w}^{\mathsf{T}}\mathbf{s})$ by the properties of $\sigma(.)$. Furthermore, the quantity $\mathbf{w}^{\mathsf{T}}\mathbf{s}$ has an interpretation as the *log odds*, i.e. $\ln[p(y = 1|\mathbf{s})/p(y = 0|\mathbf{s})]$ [7], as one can verify by straightforward algebra.

Using the development above, we are now able to write the likelihood function for the logistic regression model [7]:

$$p(\mathbf{y}|\mathbf{w}) = \prod_{n=1}^{N_{\mathrm{dev}}} \left\{ \sigma(\mathbf{w}^{\mathsf{T}}\mathbf{s})^{y_n} \sigma(-\mathbf{w}^{\mathsf{T}}\mathbf{s}))^{1-y_n} \right\}. \quad (2)$$

Maximum likelihood (ML) estimate of $\mathbf{w}$ can be found by taking the negative logarithm of (2), which yields the *cross-entropy* cost [7]:

$$-\sum_{n=1}^{N_{\mathrm{dev}}} \left\{ y_n \ln \sigma(\mathbf{w}^{\mathsf{T}}\mathbf{s}_n) + (1 - y_n) \ln \sigma(-\mathbf{w}^{\mathsf{T}}\mathbf{s}_n) \right\}. \quad (3)$$

[1] http://www.itl.nist.gov/iad/mig/tests/spk/

This is also known as the C_{llr} cost [22] in the speaker verification context. Minimum of (3) does not have closed form soluton [7], however it is convex, so iterative gradient descent methods can then be used to find the optimal $\mathbf{w}$.

The above formulation assumes that costs of miss and false alarm are equal ($C_{\text{miss}} = C_{\text{fa}}$) and that $P_{\text{tar}} = 0.5$. To re-calibrate the model according to the pre-specified cost parameters (C_{miss}, C_{fa} and P_{tar}), the following modification is required [22]:

$$p(y = 1|\mathbf{s}) = \sigma(\mathbf{w}^{\mathsf{T}}\mathbf{s} + \operatorname{logit} P_{\text{eff}}), \tag{4}$$

where P_{eff} is known as *effective prior*, which summarizes the three application dependent parameters into a single parameter, $P_{\text{eff}} = \operatorname{logit}^{-1}(\operatorname{logit}(P_{\text{tar}}) + \log(C_{\text{miss}}/C_{\text{fa}}))$, with $\operatorname{logit} P = \log(P/(1-P)) = -\theta$. Bayes-optimal decision is then achieved by placing the threshold to $-\operatorname{logit}(P_{\text{eff}})$.

In addition to DCF parameters, the ratio of the positive and negative examples in the development set might be highly imbalanced. This is the case with the bi-annual NIST evaluations. As an example, in the female itv-itv condition in NIST SRE 2010 only 3.45% of the trials are target (positive) trials. This would mean that the cross-entropy objective (3) will be strongly dominated by the nontarget scores ($y_i = 0$) leading to biased weights. To take this class imbalance problem into account, the cost was modified in [9] as follows:

$$\begin{aligned} C_{\text{wlr}}(\mathbf{w}, \mathbf{s}) &= \frac{P_{\text{eff}}}{N_t} \sum_{i=1}^{N_t} \log\left(1 + e^{-\mathbf{w}^{\mathsf{T}}\mathbf{s}_i - \operatorname{logit} P}\right) \\ &+ \frac{1 - P_{\text{eff}}}{N_f} \sum_{j=1}^{N_f} \log\left(1 + e^{\mathbf{w}^{\mathsf{T}}\mathbf{s}_j + \operatorname{logit} P}\right), \end{aligned} \tag{5}$$

where the two sums go through the N_t target score vectors $\mathbf{s}_i$ and the N_f non-target score vectors $\mathbf{s}_j$, respectively.

C. Score pre-warping

Since the raw base classifier scores may have different interpretations (e.g. log-likelihood ratios, SVM scores or i-vector cosine distances) with considerable variation in their scales, it is important to properly align the score distributions [23]. Note that the base classifiers typically include their internal score normalization such as T-norm [24], used for normalizing the classifier outputs across varying test segments and speakers with the help of external cohort models. Here the concern is to make global score alignment at the classifier level. To avoid confusion with speaker score normalization techniques, we refer to global classifier-level score pre-processing as *score warping*.

Most common warping is *mean and variance normalization* (MVN), also known as z-normalization. Mean (μ) and standard deviation (σ) of the entire score distribution, is estimated from the training data and applied to the held out score (s) as $s \mapsto (s-\mu)/\sigma$. MVN defines affine score normalization whose parameters can also be discriminatively learned. Attaching the class labels y to the score vectors one can find optimal linear warping parameters in terms of the cross-entropy criterion (3). We call this method as *unclipped z-calibration*. However, as the method to learn the fusion weights also optimizes the same cost function, it is expected that optimizing fusion weights without pre-warping by (5) will automatically learn the unclipped z-calibration of the scores.

TABLE I: Score warping methods used in this study.

Type of warping	No. params.	Discr. Training?
MVN	2	No
Z-cal (unclipped)	2	Yes
Z-cal (clipped)	4	Yes
S-cal	4	Yes

In addition to the previously mentioned variants where the range of the warping function was unbounded, we also consider *z-cal* [25] and *s-cal* [9] methods that intentionally set upper and lower limits on the scores. We thus call these methods *clipped* variants. These methods were originally devised to overcome the problem of labeling errors, assumption being that small portion of target trials were accidentally marked as non-target and vice versa. In [9], s-cal was applied on the fusion *output* but we apply it to the *inputs* before fusion. By introducing clipping in the score warping, non-linearity is applied. This leads to a score warping effect that fusion training is not able to recreate.

Both z-cal and s-cal aim at converting arbitrary scores to well-calibrated log-likelihood ratios (LLRs). The s-cal warping is,

$$\text{LLR}_{\text{scal}}(s) = \log \frac{(\operatorname{logit}^{-1}\alpha)(e^{xs+y} - 1) + 1}{(\operatorname{logit}^{-1}\beta)(e^{xs+y} - 1) + 1}, \tag{6}$$

where the saturation parameters α, β and the affine parameters x and y are optimized using the development set, with the attached ground truth labels so that the C_{llr} cost in (3) is optimized [22]. As the problem is no more convex in the unknowns, we utilize unconstrained nonlinear Nelder-Mead optimization algorithm to find locally optimum values for α, β, x, y. In each new estimate of the parameters, the development set scores are warped using (6) and C_{llr} in (3) computed which the optimizer uses for local optimization; we utilize Matlab's `fminsearch` function to implement this. Occassionally the optimizer produced singular solutions. Those were detected, by noticing that then C_{llr} is one, and rejected. If a solution was rejected then new one is computed by stronger regularization.

The z-cal warping function is defined similarly to s-cal, only difference being that instead of smooth sigmoidal shape, z-cal defines a piece-wise linear function with hard thresholding (clipping). Z-cal is defined as:

$$\text{LLR}_{\text{zcal}}(s) = (s - x_{\min})\frac{y_{\max} - y_{\min}}{x_{\max} - x_{\min}} + y_{\min}, \tag{7}$$

where we set $\text{LLR}_{\text{Zcal}}(s) = y_{\min}$ for all scores satisfying $\text{LLR}_{\text{zcal}}(s) < y_{\min}$; similarly $\text{LLR}_{\text{zcal}}(s) = y_{\max}$ for all scores with $\text{LLR}_{\text{zcal}}(s) > y_{\max}$. Z-cal parameters are optimized in a same was as S-cal parameters. Score warping methods selected for this study have been summarized in Table I.

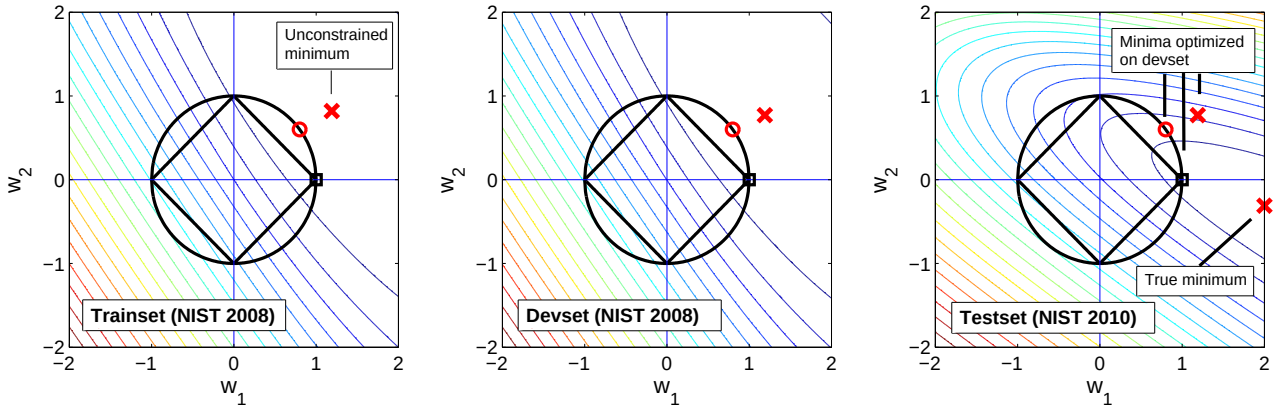


Fig. 1: Visual intuition behind sparse classifier fusion. We display the contours of the C_{wlr} cost function for score fusion of two classifiers. In each panel, the global minimum of C_{wlr} is indicated by a red cross. In the constrained optimization case, we search for the minimum instead inside a constraint region specified by $(w_1^p + w_2^p)^{(1/p)} \leq 1$ (here, the cases $p = 1$ and $p = 2$ are visualized). As can be seen, the case $p = 1$ finds a *sparse* solution in the sense that classifier 2 is zeroed out. Luckily this solution hits closest to the true minimum on the unseen test data. Even L_2 regularization ($p = 2$) gives smaller cost than the unconstrained solution, suggesting that regularization and sparsification might be particularly useful for classifier fusion under unpredictable data mismatches.

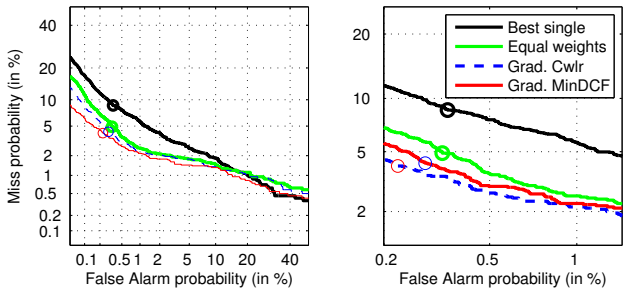


Fig. 2: Comparison of fusion methods using the full ensemble S-cal warping on Trainset. The best individual classifier (for ActDCF) is also shown. The circles indicate the ActDCF points.

III. Sparse Classifier Fusion

In this study, we propose to use ℓ_0 and ℓ_1 regularization to obtain sparse $\mathbf{w}$. It is then natural to ask whether unregularized logistic regression (i.e. maximum likelihood training) can produce a sparse solution.

Maximum likelihood solution of the $\mathbf{w}$ can be characterized as [21]:

$$\mathbf{w} = \mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0), \tag{8}$$

if class-conditional densities follow Gaussian distribution. In (8), $\mathbf{\Sigma}$ is the shared covariance matrix and μ_1 and μ_0 are the class-conditional mean vectors. If we take $\mathbf{\Sigma}$ to be diagonal, as was assumed in [8], then for one base classifier,

$$w_i = \frac{1}{\sigma_i} d_i, \tag{9}$$

where d_i is the difference between the means. It is clear that, under these assumptions, w_i can be exactly zero only when means of target and non-target score distributions completely match.

A. Classifier Ensemble Selection as Regularization

Up to this point, we have defined the standard fusion framework, assuming a full ensemble of L classifiers. Now, instead of optimizing the weights in the L-dimensional space, we are in search of both the optimal classifier ensemble and weights. For the moment, assume that we have decided on an appropriate size of the ensemble given by integer variable K, $K < L$. We would like to minimize (5) subject to this constraint. Obviously, one can simply enumerate all the $\binom{L}{K} = \frac{L!}{K!(L-K)!}$ possible classifier ensembles to ensure that the size constraint is satisfied, and optimize the weights by minimizing C_{wlr} for each of these ensemble candidates and choosing the one that minimizes the cost function. In pseudocode, of the subset selection method is presented in Algorithm 1.

Algorithm 1 Joint ensemble selection and fusion training.

1: Given: labeled scores from L classifiers and required ensemble size $K \leq L$
2: Let $\mathcal{C} = \{\mathbf{c} \in \{0,1\}^L; |\mathbf{c}| \leq K\}$ contain all the binary masks with exactly K nonzero elements
3: **for each** mask $\mathbf{c}_n \in \mathcal{B}$ **do**
4: Find optimal weights $\mathbf{w}^*(n)$ using Trainset by minimizing (5) on scores corresponding to classifiers with nonzero index in $\mathbf{c}_n$. Record the corresponding value $C_{\text{wlr}}(n)$.
5: **end for**
6: Return $\{\mathbf{c}(n^*), \mathbf{w}(n^*)\}$, where $n^* = \arg\min_n C_{\text{wlr}}(n)$ for Evalset.

Interestingly, this can also be casted in a regularization framework as follows. First, note that the ℓ_0-norm of vector

$\mathbf{w}$, defined as [14] $\|\mathbf{w}\|_0 \triangleq \sum_i |w_i|^0$, counts the number of nonzero elements in $\mathbf{w}$. This is because $|w_i|^0$ equals 1 everywhere except for $w_i = 0$ one defines it as 0. Thus, an equivalent formulation of the above combinatorial optimization problem is

$$\min_{\mathbf{w}} \; C_{\text{wlr}}(\mathbf{w}, \mathcal{S}) \quad \text{s.t.} \quad \|\mathbf{w}\|_0 \leq K, \tag{10}$$

But the minimizer of (10) is the same as that of

$$\min_{\mathbf{w}} \; \{C_{\text{wlr}}(\mathbf{w}, \mathcal{S}) + \lambda \|\mathbf{w}\|_0\}, \tag{11}$$

since the function in (11) is the Lagrangian of the constrained optimization problem in (10) [26]. This merely states that Algorithm 1 is equivalent with (11) but does not change the complexity of the problem

Although it is clear that that the combinatorial search outlined above is not very practical for large L, it is guaranteed to give the optimum classifier ensemble choice for a given set of data. In this study, we find it a useful analysis tool for studying the generalization properties of other sparsity-promoting fusion schemes given below; analogous to the common use of the difference ActDCF - MinDCF as a measure of calibration error, we can determine the best *realizable* classifier ensemble (found from training set) and compare the result to the best achievable *oracle* result on the evaluation set.

B. Practical Regularization via Ridge, LASSO and Elastic Net

For computational reasons $\|\mathbf{w}\|_0$ constraint is typically, approximated as $\|\mathbf{w}\|_1$ constraint [14]. This constraint is known as LASSO. Norm can also be constrained by $\|\mathbf{w}\|_2^2$, which corresponds then to *ridge regression*, however ridge constraint is not an sparsity promoting constraint whereas LASSO is [14].

In the case of LASSO constraint, (10) takes the following form:

$$\min_{\mathbf{w}} \; C_{\text{wlr}}(\mathbf{w}, \mathcal{S}) \quad \text{s.t.} \quad \|\mathbf{w}\|_1 \leq t. \tag{12}$$

Then the Lagrange coefficients will give us,

$$\min_{\mathbf{w}} \; \{C_{\text{wlr}}(\mathbf{w}, \mathcal{S}) + \lambda \|\mathbf{w}\|_1\}. \tag{13}$$

It is known that the larger λ, the more norm $\|\mathbf{w}\|$ will be shrunk [18]. Example of (13) on real data can be seen in Fig. 1, where two base classifiers are fused. From the example it is clear that weights found by the direct optimization of (5) would lead to non-optimal solution for the NIST SRE 2010 data set.

If the optimization is based on the Eq. (13), then the correspondence between λ and shrinkage threshold t can be found by a binary search on possible λ values. In each iteration we select one λ value and optimize weights using it, output is then the norm of the weights. Final weight vector is the one which norm is closest to the target t, but does not violate it.

Elastic-net, on the other hand, is based on the idea that we can combine both regularizers into one constraint optimization problem,

$$C_{\text{wlr}}(\mathbf{w}, \mathcal{S}) + \lambda \left(\alpha \|\mathbf{w}\|_1 + (1-\alpha)\|\mathbf{w}\|_2^2\right). \tag{14}$$

TABLE II: Selection of the three datasets used in this study. We focus on the core-condition itv-itv subset with female trials.

Dataset	Usage	Data source	# Trials
Trainset	Train fusion parameters	NIST 2008 itv-itv ♀subset	2434 t, 238971 f
Evalset 1	Compare fusion and warping methods and classif. selection	NIST 2008 itv-itv ♀subset	2408 t, 239244 f
Evalset 2	Validate results	NIST 2010 itv-itv ♀subset	5235 t, 146623 f

As can be seen, Eq. (14) is a generalized variant of both LASSO and ridge regression, we can always find such a α where, in terms of performance, elastic-net will at least not lose to LASSO or ridge regression. However, whereas LASSO and ridge regression had to select only one regression parameter, now we need to crossvalidate over a 2-d space. In this work we use the methodology, where α parameter is first fixed and then shrinkage factor can be cross validated as in LASSO and ridge. In practice, α will also be cross validated in such a way that best α and shrinkage factor will be selected based on cross validation set to be applied on the evaluation set.

Depending on the chosen regularization method, there are different strategies to optimize (13). Since logistic regression using quadratic regularization is differentiable, it can be efficiently optimized using standard packages [7]. Situation is not so simple for LASSO regularization. In [18], a *quadratic programming* (QP) solution was proposed to it by rewriting the constraints in (13) to a more convenient form. However, more recent techniques are faster in practice, for that reason we apply *projectionL1* algorithm [27] that optimizes the Lagrangian form Eq. (13). We apply the same method to elastic-net, as, sum of two convex functions is still convex, we can minimize $C_{\text{wlr}}(\mathbf{w}, \mathcal{S}) + \lambda(1-\alpha)\|\mathbf{w}\|_2^2$, given $\lambda\alpha\|\mathbf{w}\|_1$ as the constraint.

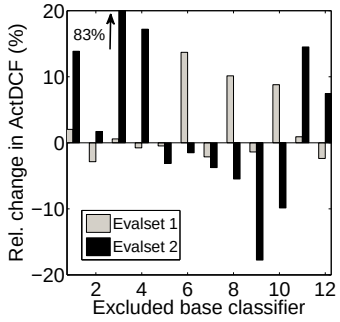


Fig. 3: Excluding the base classifier one by one from the full ensemble (relative change in ActDCF in comparison to full ensemble).

TABLE III: Twelve base classifiers, calibrated using MVN, are constructed using different cepstral features and speaker modeling techniques.

	Classifier	Feature	Evalset 1				Evalset 2			
			EER (%)	MinDCF ($\times$100)	ActDCF ($\times$100)	ActDCF-MinDCF	EER (%)	MinDCF ($\times$100)	ActDCF ($\times$100)	ActDCF-MinDCF
1	GMM-UBM-JFA	PLP	3.44	1.6748	1.6979	0.0231	7.18	3.3108	3.3911	0.0803
2	GMM-UBM-JFA	PLP	3.45	1.4309	1.5547	0.1238	5.74	2.3852	2.4268	0.0416
3	GMM-UBM-JFA	PLP	3.32	1.4760	7.7305	6.2545	4.62	2.6668	8.2292	5.5624
4	GMM-UBM-JFA	LPCC	3.99	1.9056	7.8119	5.9063	10.68	5.7845	6.5031	0.7186
5	GMM-SVM-KL	PLP	3.74	1.8597	5.2105	3.3508	6.82	2.9659	6.9683	4.0023
6	GMM-SVM-KL	MFCC	3.16	1.1564	1.4921	0.3357	5.45	2.7169	2.7338	0.0168
7	GMM-SVM-KL	LPCC	3.53	1.4877	1.8412	0.3535	8.35	4.1369	6.2928	2.1559
8	GMM-SVM-KL	MLF [28]	2.95	1.2965	1.7472	0.4508	8.29	3.9229	4.4433	0.5204
9	GMM-SVM-KL	LPCC	3.82	1.9267	5.2591	3.3324	10.55	4.9308	4.9947	0.0639
10	GMM-SVM-KL	SWLP [29]	6.69	3.6348	3.6585	0.0237	10.75	5.0897	5.7239	0.6342
11	GMM-SVM-FT [30]	PLP	4.45	1.9574	6.6046	4.6472	8.60	3.7126	8.0517	4.3391
12	GMM-SVM-BHAT [31]	PLP	3.12	1.2151	1.3090	0.0938	6.28	2.9944	3.0175	0.0232

IV. Corpora, Metrics and Base Classifiers

We utilize the two most recent NIST SRE corpora, namely, NIST 2008 and NIST 2010, in our experiments[2]. The usage of each corpus is shown in Table II. To avoid any evaluation bias from pooling of incompatible subcondition scores (see [32]), we mostly focus on the female trials[3] of the interview-interview (itv-itv) sub-condition in the core task. Nevertheless, both genders and three other sub-conditions (itv-tel, mic-mic, tel-tel) are included into the final validation. The audio files from all NIST 2008 speakers were split into two disjoint parts. In this regard, audio files (including both training and test files in the official NIST 2008 SRE dataset) from the same speaker were grouped together based on the metadata available. We then split the speakers into two groups, consisting of 475 and 711 speakers, respectively. Trials were then generated separately from those two sets by assigning training and test files randomly based on the speaker information. We kept the empirical $P_{\rm tar}$ similar to the official NIST 2008 SRE trial lists. The first part, *Trainset*, is used for training the score warping parameters (S-cal was used as precalibration method), fusion weights and bias. The second part, *Evalset 1*, is used to estimate shrinkage parameter (λ) and tradeoff between LASSO and Elastic-Net (α). Parameters are then applied to the NIST SRE 2010 trials (*Evalset 2*), which serves as the evaluation purposes. For the oracle systems, the classifier ensembles are directly optimized on Evalset 2.

For evaluation of the methods, we consider the detection cost function in (1), where the cost parameters are $C_{\rm miss} = 10$, $C_{\rm fa} = 1$ and $P_{\rm tar} = 0.01$. We measure both the minimum DCF (MinDCF) and the actual DCF (ActDCF). We also consider *calibration error*, defined as the difference of ActDCF and MinDCF, and the well-known *equal error rate* (EER), corresponding to the case of equal miss and false alarm rates[4]. Unless otherwise mentioned, the classifier ensemble that yields the smallest ActDCF, on the set of interest, is selected (note that ActDCF $\triangleq$ MinDCF on the training data).

Table III shows our twelve base classifiers based on different cepstral features and four different classifiers. When subsystems share the same classifier and features, it means that the systems are independent implementations. For classifiers, we use the generative GMM-UBM-JFA [4] and the discriminative GMM-SVM approaches with KL-divergence kernel [33] and the recently proposed Bhattacharyya kernel (BHAT) [31]. We also include another recent method, feature transformation (FT) [30], as an alternative supervector for SVM. All of the methods are grounded on the universal background model (UBM) paradigm [2] and share similar form of subspace channel compensation, though the training methods differ. We used data from the NIST 2004, 2005 and 2006 corpora to train the UBM and the session variability subspaces, and additional data from the Switchboard corpus to train the speaker-variability subspace for the JFA systems. Each base classifier has its own score normalization prior to score warping and fusion. To this end, we use T-norm and Z-norm [24] with NIST 2004 and NIST 2005 data as the background and cohort training data.

To get an idea of the base classifier dependencies, Table IV shows their pairwise Pearson's correlation coefficients (r) for both the genuine and impostor trial scores on Evalset 2. The genuine and impostor correlations are shown in the upper and lower triangles respectively. The average correlation of each classifier to the rest 11 classifiers is also indicated. Because of similar short-term spectral classifiers, the correlations are generally moderate as might be expected. The maximum pairwise correlations are $r = 0.86$ and $r = 0.83$ for the genuine and the impostor trials, respectively. The corresponding minima are $r = 0.49$ and $r = 0.39$. The highest average correlations are $r = 0.75$ (genuine; classifiers $\{6,11,12\}$) and $r = 0.62$

[2]http://www.itl.nist.gov/iad/mig//tests/sre/

[3]Female trials are somewhat more difficult than males. Similar rationale was taken, for instance, in [4].

[4]For discrete data, one does not find $P_{\rm miss} = P_{\rm fa}$ exactly. In this study, we use linear interpolation between the two closest discrete data points to compute EER. For the interested reader we point to the alternative method using convex hulls on ROC curve (ROCCH), which is available in BOSARIS [12].

TABLE IV: Pairwise correlations (Pearson's r) of the 12 base classifiers on Evalset 2 for genuine (upper triangle, green) and impostor (lower triangle, red) trials. For each classifier, average r against the rest 11 classifiers are also shown.

Correlation of impostor trials	Correlation of genuine trials												Avg.	
													gen.	imp.
	1	.85	.60	.54	.71	.69	.62	.73	.67	.65	.77	.70	.68	.55
	.83	2	.67	.54	.65	.65	.60	.68	.62	.63	.75	.68	.67	.56
	.63	.73	3	.69	.62	.59	.53	.56	.55	.58	.62	.64	.60	.55
	.48	.53	.63	4	.49	.58	.61	.54	.66	.59	.54	.55	.58	.55
	.46	.48	.56	.39	5	.74	.64	.73	.82	.74	.77	.82	.70	.49
	.54	.53	.54	.53	.53	6	.86	.85	.76	.86	.81	.85	.75	.59
	.48	.47	.45	.70	.41	.68	7	.80	.75	.86	.75	.79	.71	.58
	.56	.52	.45	.51	.44	.67	.65	8	.75	.82	.86	.82	.74	.57
	.39	.40	.44	.73	.47	.55	.72	.55	9	.80	.74	.77	.72	.53
	.47	.46	.49	.57	.48	.66	.70	.61	.63	10	.79	.83	.74	.56
	.62	.59	.48	.40	.51	.59	.51	.63	.42	.49	11	.82	.75	.54
	.58	.58	.61	.55	.65	.70	.66	.65	.59	.65	.64	12	.75	.62

(impostor; classifier 12). The systems are less correlated for impostor trials.

V. Results

A. Choosing Score Warping and Fusion Training Methods

We first compare the score warping and fusion training methods on the full set of $L = 12$ base classifiers in Table V. The first three rows display the best individual classifier for each error metric without any score warping (EER, MinDCF, ActDCF) and the last column shows the calibration error. First three rows show best individual base classifiers in terms of (ActDCF, MinDCF and EER), we notice that as scores are not pre-calibrated calibration error is quite large.

As expected, fusion improves accuracy over the best single classifier systematically. Regarding score warping, Z-cal and S-cal yield very similar and lower errors compared to non-clipping score warping and without warping. It is interesting to note that fusion training where objective is weighted cross-entropy with no-calibration or linear calibration yields slightly different MinDCFs. In addition, generative warping strategy by MVN also yields different but comparable results to all three non-clipping variants.

Comparing the fusion training methods, the gradient C_{wlr} systematically outperforms the other methods in all three costs. The DET plot in Fig. 2 confirms this. We find that direct optimization of MinDCF produces generally higher error rates than logistic regression (C_{wlr}) which does only indirect minimization. This suggests that logistic regression offers better generalization performance. For the rest of the experiments, we choose gradient C_{wlr} with S-cal.

B. Impact of Each Base Classifier

How much each base classifier contributes to the full ensemble accuracy? We measure this by excluding each classifier from the full ensemble one by one and comparing the relative change to the full ensemble. The relative ActDCF changes, using gradient C_{wlr} with S-cal, are shown in Fig. 3 for both Evalsets. For a particular classifier, positive value indicates its usefulness in the fusion pool (as ActDCF would increase) whereas negative value suggest that excluding it improves accuracy.

Classifiers 1, 3 and 11 are useful on both corpora. According to Table III these all share the same acoustic PLP front-end and the first two share the same back-end (GMM-UBM-JFA). The most striking case is that of classifier 3 whose exclusion in Evalset 2 increases ActDCF by more than 80 % relative to full ensemble. Table III reveals that it is the system with smallest EER on Evalset 2. The classifiers 5, 7 and 9, in turn, contribute negatively on both corpora. These share the same GMM-SVM-KL back-end but one of them shares also the PLP front-end. Finally, the rest of the classifiers (2, 5, 6, 8, 10 and 12) do not behave consistently across the two corpora.

C. Effect of Ensemble Size with Oracle Bounds

Up to this point, we have studied the accuracies of the base classifier and the full ensemble. We now turn to classifier ensemble selection by using the gradient C_{wlr} method with S-cal as before. We first study the effect of ensemble size on Evalset 2. For a given ensemble size, K, and error metric (ActDCF, MinDCF, EER), we consider three summary values out from the $\binom{12}{K}$ combinations in Fig. 4. *Best real ensemble* is the optimum ensemble, where fusion device is trained on Trainset and subset is selected using Evalset 1 and evaluated on Evalset 2 whereas *best oracle ensemble* is computed by direct optimization on the Evalset 2, by using the key file. Note that the oracle considers ensemble selection only – the fusion weights of each candidate ensemble are trained from the training set (we will return to this in the next subsection). Finally, *worst oracle ensemble* is the worst ensemble choice on the given Evalset and gives an idea how bad the result can be with unlucky ensemble selection.

TABLE V: Fusion of all the $L = 12$ base classifiers on the Evalset 1. The first three rows show the individually best base classifiers.

Fusion method	Score warping	EER	MinDCF	ActDCF	ActDCF-MinDCF
Best ActDCF	–	3.74	1.8597	**3.0131**	**1.1534**
Best MinDCF	–	3.16	**1.1564**	18.4600	16.600
Best EER	–	**2.95**	1.2965	14.7607	13.464
Equal weights	–	2.09	0.8385	5.9863	5.1478
	MVN	2.10	0.8219	2.3085	1.4865
	Linear	2.08	0.8080	1.1022	0.2942
	S-cal	**2.03**	0.7907	**0.9176**	**0.1269**
	Z-cal	1.99	**0.7786**	0.9617	0.1830
Grad. C_{wlr}	–	1.83	0.6172	0.6231	**0.0059**
	MVN	1.83	0.6139	0.6235	0.0096
	Linear	1.83	0.6135	0.6231	0.0096
	S-cal	1.70	0.6031	**0.6147**	0.0116
	Z-cal	**1.66**	**0.5940**	0.6183	0.0243
Grad. MinDCF	–	2.03	0.7038	2.1931	1.4892
	MVN	2.03	0.7095	4.2973	3.5878
	Linear	2.03	0.7159	**1.5044**	**0.7885**
	S-cal	**1.89**	**0.6440**	2.7454	2.1014
	Z-cal	1.95	0.6631	9.9502	9.2871

Is there any practical advantage of using a subset ensemble of classifiers instead of the full ensemble? We see

that ensemble sizes 6-9 bring real benefit in contrast to full ensemble. Smaller subset sizes, from 2-5, shows unpredictable behaviour on the real system. As is expected, predicted real system does not reach oracle performance, showing that even more improvement can be obtained by better prediction of the subset.

Another interesting observation is that, for all the three error metrics, the empirical bounds on best and worst performance approach each other for increased ensemble size. This can be understood by noting that $\binom{12}{K}$ is small when K is close to 12; thus, there are simply less choices to make bad (or good) classifier selection. Being more fair and comparing ensemble pools of *equal size*, say, $\binom{12}{2} = \binom{12}{10} = 66$ or $\binom{12}{3} = \binom{12}{9} = 220$, it is clear from Fig. 4 that using the larger ensemble leads to more stable fusion, which is intuitively reasonable. Averaging similar spectral system scores helps in reducing variance (uncertainty) of the fused score as discussed in [34], [35].

Table VI summarizes the subset selection cross-validation results on Evalset 2. Real system performance is contrasted against oracle selection rule. In the table best individual ("best ind.") and full ensemble ("full ens.") accuracies are also indicated, and as before, *best* means best in ActDCF.

As earlier, the results in Table VI are divided into *real* and *oracle* performances. *Subset ensemble* ("subset ens.") refers to the best joint ensemble selection and fusion choice, out from all the $2^{12} - 1 = 4095$ possibilities. For the real and oracle systems, the ensemble selection is carried out on Evalset 1 and Evalset 2, respectively. Oracle ensemble results are similar to Fig.4, where the weights of the 4095 ensemble candidates are optimized on Trainset, and the oracle selects the best ensemble on the Evalset 2.

We make the following observations from Table VI:

- Realistic subset ensemble has better performance on all the three metrics over full ensemble. Relative improvement is 4.4% for EER, 2.8% for MinDCF and 8.7% for ActDCF.
- On the oracle scenario, subset ensemble fusion gives smaller error rates using smaller ensemble size (5) than predicted ensemble (7).
- For the subset ensemble, improvement in EER from the real to oracle system is larger than for MinDCF, relative reduction of 10% and 6.5% respectively.
- Subset selected by oracle contains classifiers {2,3,5,6} which have low EER (Table III). For the classifier 10 with EER of 10.75 % it is less clear why it is included in the oracle set. The correlations in Table IV do not reveal this either.
- In the real systems, all three error metrics are reduced by fusion, using both subset and full ensemble methods, over the best individual system. This confirms that fusing even a rather homogenous set of low-level spectral classifiers is beneficial.

D. Extended Results on Other Conditions

Throughout the experiments, we have restricted ourselves to female trials of the interview-interview subcondition. To validate our observations on a broader set of conditions, Table VIII shows accuracies for other selected subconditions of the NIST 2010 core task, for only female trials. We consider four subconditions that correspond to pooled data from the NIST 2010 common conditions as listed in Table VII.

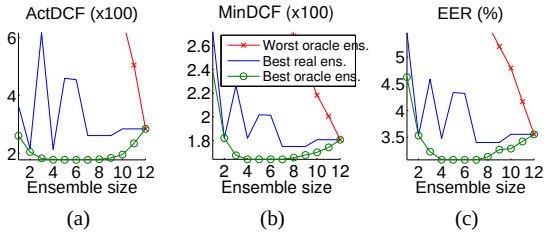
All fusion strategies – best individual, full ensemble and regularized variants – are in non-oracle setting, that is, fusion training are carried out on a training set and regularization parameters are estimated using Evalset 1. In this experiment, we also make sure that NIST 2008 training set and NIST 2010 evaluation set are condition-matched under each subcondition.

In addition to ActDCF, MinDCF and EER, we report the actual number of misses (N_{miss}) and false alarms (N_{fa}) at each operating point. Further, we carried out McNemar's significance testing [36], [37] on both types of errors at 95 % confidence level, to find when they differ between the subset and full ensemble fusions. The significantly differing cases (from the full ensemble) are denoted by $\star$.

In Table VIII, we show the recognition results for different NIST SRE 2010 sub-conditions (itv-itv, itv-tel, mic-mic and tel-tel). Here, baseline method refers to the unregularized

TABLE VI: Comparing different ensemble selections on Evalset 2. The best individual classifier (best for ActDCF) is selected using Evalset 1 (*real* systems) or Evalset 2 (*oracle* systems).

	Fusion	Weights training	Ensemble selection	Included classifiers	EER (%)	MinDCF (×100)	ActDCF (×100)
Real	Best ind.	–	Evalset 1	{6}	5.45	2.7169	3.6453
	Subset ens.	Trainset	Evalset 1	{1,2,3,4,5,6,9}	**3.40**	**1.7506**	**2.6119**
	Full ens.	Trainset	–	all	3.55	1.8072	2.8420
Orcl.	Best ind.	–	Evalset 2	{2}	5.74	2.3852	2.6063
	Subset ens.	Trainset	Evalset 2	{2,3,5,6,10}	3.08	1.6426	1.7747



(a) (b) (c)

Fig. 4: Effect of ensemble size to accuracy (Evalset 2). For a fixed ensemble size (K), the lowest (green) and highest (red) lines show the best and worst possible selections out from the $\binom{12}{K}$ choices from Evalset 2 (NIST SRE 2010). The middle (blue) line indicates the actual ensemble selected by cross-validation Evalset 1.

TABLE VII: Chosen NIST 2010 subconditions.

	NIST 2010 common cond.								
	1	2	3	4	5	6	7	8	9
itv-itv	×	×							
itv-tel			×						
mic-mic							×		×
tel-tel					×	×		×	

TABLE VIII: Comparison of fusion methods for NIST SRE 2010 set, all tuning parameters have been cross validated using NIST SRE 2008 development set.

	Training method	**EER (%)**	**MinDCF (×100)**	**ActDCF (×100)**	$\frac{\|\mathbf{w}_{\text{reg}}\|_1}{\|\mathbf{w}\|_1}$	Ensemble size
itv-itv	Best Single (GSV-MFCC)	5.45	2.72	3.65		1
	no regularization	3.55	1.81	2.84	1	12
	subset sel.	3.40	1.75	2.61		6
	ridge	3.40	1.70	2.51	0.96	12
	LASSO	**3.33**	**1.69**	**2.23**	0.96	6
	E-net $\alpha = 0$	3.40	1.70	2.50	0.96	12
itv-tel	Best Single (JFA-PLP)	3.03	1.39	1.75		1
	no regularization	2.40	0.98	1.74	1.0	12
	subset sel.	2.31	1.06	**1.34**		7
	ridge	2.40	**0.97**	1.65	0.86	12
	LASSO	2.40	0.99	1.63	0.71	8
	E-net $\alpha = 0.7$	2.37	**0.97**	1.47	0.66	10
mic-mic	Best Single (JFA-PLP)	6.52	3.04	3.14		1
	no regularization	5.10	2.35	4.14	1.0	12
	subset sel.	**4.80**	**2.30**	3.08		8
	ridge	5.10	**2.30**	3.04	0.66	12
	LASSO	5.62	2.44	3.23	0.56	3
	E-net $\alpha = 0.7$	4.82	**2.30**	**3.03**	0.51	6
tel-tel	Best Single (JFA-PLP)	3.62	1.58	1.74		1
	no regularization	2.33	1.12	**1.18**	1.0	12
	subset sel.	2.43	1.25	1.27		6
	ridge	2.33	1.14	1.28	0.91	12
	LASSO	**2.25**	1.19	1.27	0.91	5
	E-net $\alpha = 0.1$	2.42	1.15	1.32	0.81	12

solution (i.e. $\lambda = 0$), equivalent to the implementation of the FoCal toolkit. Best single classifier is selected based on the performance on the cross validation set, so all the methods are directly, and fairly, comparable in Table VIII. We notice that, for the itv-tel and micmic subconditions, elastic-net and subset selection achieve similar and the best results. It is interesting to note that improvement in the ActDCF is because scores are better calibrated.

General trend, when comparing MinDCF over all conditions seems to be that there are no large differences except in the mic-mic condition where no regularization clearly fails. Differences in ActDCF are mostly the product of different calibrations. Note that the bias is not regularized.

It is interesting to note that predicting the α value using cross validation set is not a trivial problem. It is clear that in the case when either LASSO or ridge wins over elastic-net in terms of ActDCF, the prediction of α was unsuccesful. Especially interesting is the itv-itv case, where prediction gave $\alpha = 0$ (i.e. ridge) and for NIST SRE 2010, LASSO was clearly better.

Regularization, however, does not bring improvement in the tel-tel condition in terms of ActDCF. For the tel-tel condition, designers of base classifiers had a very large and extensively used corpora available for tuning up their systems. In addition, selection of data sets for the estimation of session compensation parameters is more straightforward. But the interview and microphone data conditions did not have such a wealth of material backing their classifier design. It is thus expected that regularization will hurt the classification performance in the tel-tel condition. In the other conditions, significant improvement over the baseline can be achieved by any of the regularization methods.

VI. Conclusion

We have presented a sparse regularized logistic regression score fusion for speaker verification. We tuned our system using audio data from NIST SRE 2008 corpus and evaluated using NIST SRE 2010 core test conditions. We find that sparse regularization brings improvement over unregularized variant in all other sub-conditions and measures (EER, MinDCF, ActDCF) except ActDCF in tel-tel condition.

An important conclusion from the oracle experiments is that ensemble selection gives significant accuracy boost over both the best individual and the full classifier ensemble, *if one knew how to pick the correct ensemble*. The naive strategy to pick ensemble that minimizes classification cost on the training set, does not obviously lead to good generalization, and therefore alternative off-line selection strategies that also take into account the diversity of the ensemble, are required. Alternatively, run-time classifier ensemble selection for each

speech utterance, similar to adaptable fusion using auxiliary quality measures [38], [39], [40], would be an interesting direction.

REFERENCES

[1] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: from features to supervectors," *Speech Communication*, vol. 52, no. 1, pp. 12–40, January 2010.

[2] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1, pp. 19–41, January 2000.

[3] W. Campbell, J. Campbell, D. Reynolds, E. Singer, and P. Torres-Carrasquillo, "Support vector machines for speaker and language recognition," *Computer Speech and Language*, vol. 20, no. 2-3, pp. 210–229, April 2006.

[4] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of inter-speaker variability in speaker verification," *IEEE T. Audio, Speech & Lang. Proc.*, vol. 16, no. 5, pp. 980–988, July 2008.

[5] A. Solomonoff, W. Campbell, and I. Boardman, "Advances in channel compensation for SVM speaker recognition," in *Proc. ICASSP 2005*, Philadelphia, Mar. 2005, pp. 629–632.

[6] C. P. Robert, *The Bayesian Choice: from Decision-Theoretic Motivations to Computational Implementation*, 2nd ed. Springer-Verlag, 2001.

[7] C. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer Science+Business Media, LLC, 2006.

[8] S. Pigeon, P. Druytsa, and P. Verlinde, "Applying logistic regression to the fusion of the nist'99 1-speaker submissions," *Digital Signal Processing*, vol. 10, no. 1–3, pp. 237–248, January 2000.

[9] N. Brümmer, L. Burget, J. Černocký, O. Glembek, F. Grézl, M. Karafiát, D. Leeuwen, P. Matějka, P. Schwartz, and A. Strasheim, "Fusion of heterogeneous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006," *IEEE Trans. Audio, Speech and Language Processing*, vol. 15, no. 7, pp. 2072–2084, September 2007.

[10] L. Ferrer, K. Sönmez, and E. Shriberg, "An anticorrelation kernel for subsystem training in multiple classifier systems," *J. of Machine Learning Research*, vol. 10, pp. 2079–2114, 2009.

[11] N. Brümmer, "Fusion and toolkit [software package]," WWW page, June 2011, http://sites.google.com/site/nikobrummer/focal.

[12] "Bosaris toolkit [software package]," WWW page, June 2011, https://sites.google.com/site/bosaristoolkit/.

[13] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. New York: Wiley Interscience, 2000.

[14] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning – Data Mining, Inference, and Prediction*, 2nd ed. Springer, 2008.

[15] H. Li, B. Ma, K. A. Lee, H. Sun, D. Zhu, K. C. Sim, C. H. You, R. Tong, I. Kärkkäinen, C.-L. Huang, V. Pervouchine, W. Guo, Y. Li, L. Dai, M. Nosratighods, T. Tharmarajah, J. Epps, E. Ambikairajah, E.-S. Chng, T. Schultz, and Q. Jin, "The I4U system in NIST 2008 speaker recognition evaluation," in *Proc. Int. conference on acoustics, speech, and signal processing (ICASSP 2009)*, Taipei, Taiwan, April 2009, pp. 4201–4204.

[16] F. Sedlák, T. Kinnunen, K. A. L. Ville Hautamäki, and H. Li, "Classifier subset selection and fusion for speaker verification," in *ICASSP 2011*, 2011.

[17] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Computation*, vol. 4, no. 1, pp. 1–58, January 1992.

[18] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, no. 1, pp. 267–288, 1996.

[19] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of Royal Statistical Society, Series B*, vol. 67, pp. 301–320, 2005.

[20] W. Campbell, D. Sturim, W. Shen, D. Reynolds, and J. Navratil, "The MIT-LL/IBM 2006 speaker recognition system: High-performance reduced-complexity recognition," in *Proc. ICASSP 2007*, vol. IV, 2007, pp. 217–220.

[21] M. I. Jordan, "Why the logistic function? a tutorial discussion on probabilities and neural networks," Massachusetts Institute of Technology, Cambridge, MA, Tech. Rep., August 1995.

[22] N. Brümmer and J. Preez, "Application-independent evaluation of speaker detection," *Computer Speech and Language*, vol. 20, pp. 230–275, April-July 2006.

[23] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern Recognition*, vol. 38, no. 3, pp. 2270–2285, 2005.

[24] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 42–54, January 2000.

[25] "Z-cal," 2006, http://ww.dsp.sun.ac.za/~nbrummer/focal/cllr/calibration/z_cal/index.htm. [Online]. Available: http://ww.dsp.sun.ac.za/~nbrummer/focal/cllr/calibration/z_cal/index.htm

[26] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications, 1998.

[27] M. Schmidt, G. Fung, and R. Rosales, "Fast optimization methods for L1 regularization: A comparative study and two new approaches," in *ECML 2007*, Warsaw, Poland, September 2007.

[28] C.-L. Huang, H. Su, B. Ma, and H. Li, "Speaker characterization using long-term and temporal information," in *Proc. Interspeech 2010*, Makuhari, Japan, September 2010, pp. 370–373.

[29] R. Saeidi, J. Pohjalainen, T. Kinnunen, and P. Alku, "Temporally weighted linear prediction features for tackling additive noise in speaker verification," *IEEE Sign. Proc. Lett.*, vol. 17, no. 6, pp. 599–602, 2010.

[30] D. Zhu, B. Ma, and H. Li, "Joint MAP adaptation of feature transformation and gaussian mixture model for speaker recognition," in *Proc. Int. conference on acoustics, speech, and signal processing (ICASSP 2009)*, Taipei, Taiwan, April 2009, pp. 4045–4048.

[31] C. H. You, K. A. Lee, and H. Li, "GMM-SVM kernel with a Bhattacharyya-based distance for speaker recognition," *IEEE Trans. Audio, Speech and Language Processing*, vol. 18, no. 6, pp. 1300–1312, August 2010.

[32] D. Leeuwen, "A note on performance metrics for speaker recognition using multiple conditions in an evaluation," Research note, June 2008, http://sites.google.com/site/sretools/cond-weight.pdf?attredirects=0.

[33] W. Campbell, D. Sturim, and D. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 308–311, May 2006.

[34] N. Poh and S. Bengio, "Why do multi-stream, multi-band and multi-modal approaches work on biometric user authentication tasks?" in *Proc. ICASSP 2004*, vol. 5, Montreal, Canada, May 2004, pp. 893–896.

[35] G. Brown, J. Wyatt, R. Harris, and X. Yao, "Diversity creation methods: A survey and categorisation," *Journal of Information Fusion*, vol. 6, no. 1, pp. 5–20, March 2005.

[36] D. Leeuwen, A. Martin, M. Przybocki, and J. Bouten, "NIST and NFI-TNO evaluations of automatic speaker recognition," *Computer Speech and Language*, vol. 20, pp. 128–158, April-July 2006.

[37] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: a Guide to Theory, Algorithm, and System Development*. New Jersey: Prentice-Hall, 2001.

[38] L. Ferrer, M. Graciarena, A. Zymnis, and E. Shriberg, "System combination using auxiliary information for speaker verification," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2008)*, Las Vegas, Nevada, March-April 2008, pp. 4853–4856.

[39] K. Kryszczuk, J. Richiardi, P. Prodanov, and A. Drygajlo, "Reliability-based decision fusion in multimodal biometric verification systems," *EURASIP Journal of Advances in Signal Processing*, no. 1, p. Article ID 86572, 2007.

[40] F. Huenupán, N. Yoma, C. Garretón, and C. Molina, "On-line linear combination of classifiers based on incremental information in speaker verification," *ETRI Journal*, vol. 32, no. 3, pp. 395–405, June 2010.